2012 26th International Conference on Advanced Information Networking and Applications Workshops

Bio Named Entity Recognition based on Co-training Algorithm

Tsendsuren Munkhdalai[1], Meijing Li[1], Taewook Kim[1], Oyun-Erdene Namsrai[2], Seon-phil Jeong[3], Jungpil Shin[4], Keun Ho Ryu[1,4]

[1]Database/Bioinformatics Laboratory, Chungbuk National University, Cheongju, South Korea
[1]E-mail: {tsendeemts, mjlee, twkim, khryu}@dblab.chungbuk.ac.kr
[2]School of Information Technology, National University of Mongolia, Ulaanbaatar, Mongolia
[2]E-mail: oyunerdene@num.edu.mn
[3]Division of Science & Technology, BNU-HKBU United International College, Zhuhai, China
[3]E-mail: spjeong@uic.edu.hk
[4]Multimedia Systems Lab, School of Computer Science and Engineering, The University of Aizu, Aizu-Wakamatsu, Fukushima, Japan
[4]E-mail: {jpshin, khryu}@u-aizu.ac.jp

***Abstract*—One essential task in extracting information from biomedical literature is the bio Named Entity Recognition (NER) process, which basically defines the boundaries between typical words and biomedical terminology in particular text data, and assigns them based on domain knowledge. This paper presents a semisupervised integration of completely different classifiers to cover knowledge from unlabeled data to recognize bio named entities in text. We modified the original co-training, a semisupervised learning algorithm, with a scalable feature processing schema, which extracts the bio NER feature from a number of unlabeled data and converts different types of feature sets. Our base result shows that the classifiers of co-training achieve significant learning from unlabeled data.**

Keywords-Bio named entity recognition; co-training; semisupervised learning; feature processing; text mining

I. INTRODUCTION

As biomedical literature sharply grows on servers in a semi-structured document format, bio-text mining has been intensively investigate to seek information in a more accurate and efficient manner. One essential task in such information extraction system is the bio Named Entity Recognition (NER) process, which basically defines the boundaries between typical words and biomedical terminology in particular text data, and assigns them based on the domain knowledge, where the terminology belongs to a category.

The bio NER task is not trivial, which made by several factors. NER has reached high level performance or recognition level as human in newswire domain where accuracy above 90% are prevalently informed. However, in biology domain, recognition of named entities has not so far proven to be such consistently affected by the problems that need to be solved, namely, a number of new gene names are growing continually, authors do not choose a standardized name as technical term and the technical terms, such as genes names, are occur with other terminology[1].

There are three main approaches adopted for bio NER; rule-based, dictionary based, and machine learning approaches. In the dictionary based approach, a previously prepared terminology list is matched though a given text to retrieve chunks containing the location of the terminology word. However, medical and biological text can contain a new terminology that might not be recognized by the dictionary based approach.

The rule based approach defines the particular rules by observing the general features of bio entities in biomedical text [2]. In order to identify any named entity in all text data, the rule generation process has to cover a huge amount of text to collect accurate rules. On the other hand, the rules are usually collected by domain experts requiring a lot of effort.

Recently, machine learning approaches have become dominant in bio NER, due to their unique characteristics that are well tuned with one another and even other methods, such as the dictionary based method. Since the machine learning approach has been adopted, significant progress in bio NER has been achieved with approaches like Markov Model [3], Support Vector machine (SVM) [4], [5], [6], [7], [8] Maximum Entropy Markov Model [9], and Conditional Random Field (CRF) [10], [11], [12], [13], [14].

The authors in [6] showed that the dictionary based method can be combined with machine learning by matching an example in the dataset through a dictionary to extract features, compressing particular domain knowledge. Collier et al. [7] obverse the effect of variations of character-level orthographic and Part-Of-Speech (POS) tag features, and they introduce a ranked table that contains features according to their probabilities of predicting a class. As shown in the table, the orthographic features indicate the class clearly, such as a feature named as GreekLetter has a 0.96 probability value of predicting a class.

Conditional random field, a classic sequence labeling tool in text mining and natural language processing, has successfully been used in a large number of studies on bio NER, since it takes advantage of sequence labeling. Hsu et al. [12] integrate two different directions parsing, backward and forward, CRF models selected underlying the output

978-0-7695-4652-0/12 $26.00 © 2012 IEEE
DOI 10.1109/WAINA.2012.75

score of them. The most recent study on using CRF is [13], which constructs a large dictionary from unlabeled data via a feature generalization method and combines the dictionary with a CRF tagger to improve the base CRF tagger performance.

All of those methods are based on BIO tagging format in which each example in the dataset is classified either at the beginning, inside or out of a named entity; thus, the methods, specially SVM due to its kernel nature, may suffer from the multi-label class problem. In our previous work [15], we solve the name boundary problem of named entity, which is discussed in [7], by introducing our BFSM algorithm that builds a finite state machine parser from a mixture of Bayesian networks containing a POS tag as its random variable. The BFSM analyzes the sentence structure and selects a part-of-speech of words to assemble candidates of a bio named entity, and thus the multi-class problem becomes a two-class problem as classifying a candidate word as true or false. However, the weakness of this method is that during the candidate assembling process, some examples are discarded out of the candidates because of the rare case of their POS tags. To solve this problem, we integrate two completely different classifiers, one of which classifies examples into three-class labels, while the other classifies examples into two-class labels in a co-training fashion [16], which relies on a scalable feature processing schema developed to extract bio NER features from a number of unlabeled data, and converts different types of feature sets.

The rest of this paper is organized as follows. Section 2 introduces the proposed methodology and the elements in the co-training integration. Section 3 reports the environmental settings in the integration, the results of learning improvement of the classifiers, and a discussion of the experiment. Finally, we summarize the main conclusions achieved and present our future work direction.

II. THE INTEGRATION METHODOLOGY

A. *Adopting co-training for the integration*

The fact that in most real life application domains, namely image classification, face recognition, text categorization and even named entity recognition, a good labeled data set is obtained in an expensive way; however, the fact that unlabeled data are less expensive and plentiful to obtain motivates the development of co-training, a semi-supervised learning algorithm. In this algorithm, two classifiers initially designed in two different feature sets learn via the training data and are then used for the unlabeled data, and at each round of the learning iteration, the algorithm elects the example from the unlabeled data based on the labeling confidence to learn the classifiers on them.

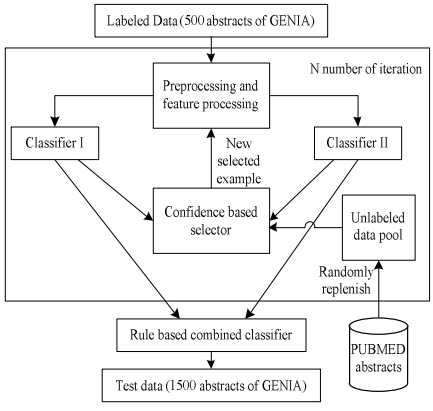


Figure 1. Flowchart of the semisupervised algorithm for two-classifier integration

Blum and Mitchell [16] proved the idea behind co-training from a mathematical view and discussed two assumptions of co-training in efficient learning:

- Let F_1 feature set be the basis of classifier C_1, F_2 feature set be basis of classifier C_2, and in co-training, F_1 and F_2 feature sets have to be conditionally independent given a class label.
- Instance distribution with target function to respect both feature sets should be compatible with each other, so that it is possible to predict the same label of the examples using each feature set independently of the other.

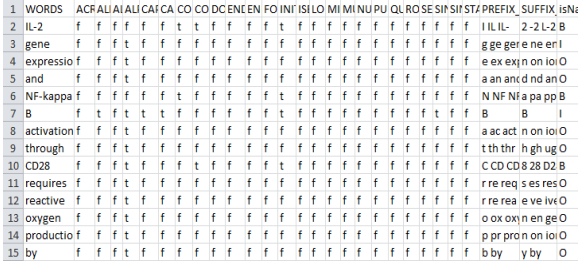
The idea of co-training is that if the assumptions hold, then the classifiers classify the example from two different views and an example classified by one is viewed as a random training example by the other one. In this case, the example achieves learning advancement of the other classifier. Even though most real-world applications rarely meet these assumptions, Nigam and Ghani [17] show that there is an advantage to be gained using co-training.

In the original co-training algorithm, it is assumed that the class labels of two classifiers are the same, which differs from our integration methodology where we convert three-class labeled examples from one classifier to two-class

TABLE I. THE CO-TRAINING ALGORITHM

Given: A small set L of training examples
A large set U of unlabeled examples
Two sets F_1 and F_2 of features, redundantly sufficient
Co-training parameters N, n and p

```
u = init_random(U); // creating an unlabeled data pool
for i = 0 to N do // loop for N iterations
    for j = 1 to 2 do
        train_classifier(Cj, Fj, L);
            // training C1, C2 classifiers based on F1, F2 feature sets
    end for
    for j = 1 to 2 do
        classify(Cj, u); // classify the unlabeled data pool
        pick_top_confident(p, u, "TRUE");
            // pick most confident p number of positive examples
        pick_top_confident(n, u, "FALSE");
            // pick most confident n number of negative examples
    end for
    update_dataset(L);
        // refresh training data L with newly picked examples
    replenish_random(u, U, 2p + 2n);
    // choose 2p + 2n examples from U to replenish
      unlabeled data pool u
end for
```



WORDS	ACR	ALI	AI	ALI	CAI	CA	CO	CO	DC	EN	EN	FO	INI	ISI	LO	MI	MI	NU	PU	QU	RO	SE	SIN	SIT	STA	PREFIX_	SUFFIX_	isNa
IL-2	f	f	f	f	f	t	t	f	f	f	f	t	f	f	f	f	f	f	f	f	f	f	f	f	f	I IL IL-	2 -2 L-2	B
gene	f	f	f	t	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	g ge ger	e ne ene	I
expressio	f	f	f	t	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	e ex exp	n on ion	O
and	f	f	f	t	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	a an and	d nd and	O
NF-kappa	f	f	f	f	f	t	f	f	f	f	f	f	t	f	f	f	f	f	f	f	f	f	f	f	f	N NF NFa	a pa pp	B
B	f	t	f	t	t	t	f	f	f	f	f	f	t	f	f	f	f	f	f	f	f	f	t	f	f	B	B	I
activation	f	f	f	t	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	a ac act	n on ion	O
through	f	f	f	t	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t th thr	h gh ugh	O
CD28	f	f	f	t	f	f	f	t	f	f	f	f	f	t	f	f	f	f	f	f	f	f	f	f	f	C CD CD	8 28 D2	B
requires	f	f	f	t	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	r re req	s es res	O
reactive	f	f	f	t	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	r re rea	e ve ive	O
oxygen	f	f	f	t	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	o ox oxy	n en gen	O
productio	f	f	f	t	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	p pr pro	n on ion	O
by	f	f	f	t	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	b by	y by	O

WORD	CONTI	CONT	GENE	GEN	GENE	GENE	GENI	GENI	LOWI	NOTO	NOTI	NOTG	NOTG	NOTG	NOTO	NOTO	NOTI	NOTG	PETE	PETI	LENG	Class
IL-2	f	f	f	t	t	f	t	f	f	f	f	f	f	f	f	f	f	f	t	f	4	TRUE
IL-2 gene	ft	ft	ff	tt	tf	ff	tf	ff	f	ft	ff	ff	ff	ff	ff	ff	ff	ft	tf	f	9	TRUE
IL-2 gene	ftf	ftt	fff	ttf	tff	fff	tff	fff	f	ftf	fff	fff	fff	fff	fff	fff	fff	ftt	tff	f	20	FALSE
gene	t	t	f	f	f	f	f	f	t	f	f	f	f	f	f	f	f	t	f	f	4	FALSE
gene expr	tf	tt	ff	ff	ff	ff	ff	f	tf	ff	ff	ff	ff	ff	ff	ff	tt	ff	f	15	FALSE
expressio	f	t	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	f	f	10	FALSE
NF-kappa	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	8	FALSE
NF-kappa	ft	ff	ff	ff	tt	ff	ft	f	ff	ff	ff	ff	ff	ff	ff	ff	ft	ft	f	10	TRUE
NF-kappa	ftf	fft	fff	fff	fff	ttf	fff	ftt	f	fff	fff	fff	fff	fft	fff	fff	fff	ftt	ftf	f	21	FALSE
B	t	f	f	f	f	f	t	f	f	f	f	f	f	f	f	f	f	t	t	f	1	FALSE
B activatic	tf	ft	ff	ff	ff	ff	tt	f	ff	ff	ff	ff	ft	ff	ff	ff	tt	tf	f	12	FALSE
activation	f	t	f	f	f	f	t	f	f	f	f	f	t	f	f	f	t	f	f	10	FALSE
CD28	f	f	t	t	t	f	t	f	f	f	f	f	f	f	f	f	f	t	f	4	TRUE
reactive o	ff	ff	ff	ff	ff	ff	tf	f	ff	ff	ff	ff	ft	ff	ff	ff	ff	ff	f	15	FALSE

Figure 2. The two independent sets of features for semisupervised learning. The feature sets from orthographic and lexicons are on the left and right sides, respectively.

TABLE II. THE ORTHOGRAPHIC TO CONTEXT FEATURE CONVERSION ALGORITHM.

Orthographic to context conversion

Given: A set $D_o = \{(x_i, y_i) \mid i = 1, 2, \ldots, k\}$ of labeled examples with orthographic features;
A set $P_o = \{(c_{iB}, c_{iI}, c_{iO})\}$ of class probability;
An empty set $D_c = \{(z_j, w_j) \mid j = 0\}$ of labeled examples with context features;
An empty set $P_c = \{(c_{jTRUE}, c_{jFALSE})\}$ of class probability;

1 **for** each labeled examples $(x_i, y_i) \in D_o$ **do**
2 **if** y_i = "B" **then**
// creating new example labeled with "TRUE"
3 $D_c \leftarrow$ create(x_i.*word*, "TRUE");
4 $P_c \leftarrow c_{iB}$;
5 **else if** y_i = "I" **then**
6 **if** y_{i-1} = "B" or y_{i-1} = "I" **then**
7 z_{j-1}.*word* = merge_words(z_{j-1}.*word*, x_i.*word*);
// a simple word merging rule
8 $c_{j-1TRUE} = \max(c_{j-1TRUE}, c_{iB|I})$;
9 **else**
10 lose (x_i, y_i) as a noise
11 **end if**
12 **else** // current example is labeled with "O"
// creating new example labeled with "FALSE"
13 $D_c \leftarrow$ create(x_i.*word*, "FALSE");
14 $P_c \leftarrow c_{iO}$;
15 **end if**
16 **end for**
17 extract_context_features(D_c);
// extracting context features from the word feature of the newly created two-class labeled examples

TABLE III. THE CONTEXT TO ORTHOGRAPHIC FEATURE CONVERSION ALGORITHM.

Context to orthographic conversion

Given: An empty set $D_c = \{(z_j, w_j) \mid j = 1, 2, \ldots, k\}$ of labeled examples with context features;
An empty set $P_c = \{(c_{jTRUE}, c_{jFALSE})\}$ of class probability;
A set $D_o = \{(x_i, y_i) \mid i = 0\}$ of labeled examples with orthographic features;
A set $P_o = \{(c_{iB}, c_{iI}, c_{iO})\}$ of class probability;

1 **for** each labeled examples $(z_j, w_j) \in D_c$ **do**
2 T $\leftarrow$ tokenize(z_j.*word*);
3 **if** w_j = "TRUE" **then**
// creating new examples labeled with "B" and "I"
4 $D_o \leftarrow$ create(T_1, "B");
5 $P_o \leftarrow c_{jTRUE}$;
6 **for** q = 2 to T.*size* **do**
7 $D_o \leftarrow$ create(T_q, "I");
8 $P_o \leftarrow c_{jTRUE}$;
9 **end for**
10 **else** // current example is labeled with "FALSE"
11 **for** q = 1 to T.*size* **do**
12 $D_o \leftarrow$ create(T_q, "O");
13 $P_o \leftarrow c_{jFALSE}$;
14 **end for**
15 **end for**
16 extract_orthographic_features(D_o);
// extracting orthographic features from the word feature of the newly created three-class labeled examples
17 Filter "B" and "I" examples labeled as "O" using BIO classifier;

labeled examples from the other, and vice versa. Therefore, we modify the co-training algorithm with a scalable feature processing schema, which extracts a feature from a number of unlabeled data and converts different types of feature sets. The final classification decision is made by the combination of the two classifiers.

With the combined classifier, we can solve the problem in our previous work, as described previously. In summary, the problem is that bio named entities are discarded by a classifier from our previous study and that the classifier based on the BIO tagging format can handle them to benefit the combined classifier.

Fig. 1 shows a flowchart of our integration methodology, and Tab. 1 outlines the co-training algorithm proposed in [16]. All steps composing the integration methodology and the co-training algorithm are going to be described in the subsequent sections.

B. Preprocessing and feature processing

Preprocessing where text data is cleaned and processed via Natural Language Processing (NLP) is a preparation task for feature processing, which extracts and converts different types of feature sets.

In the preprocessing, before the text data go through the NLP step, including sentence parsing and part-of-speech tagging, the text cleaning task removes non-informative characters and replaces the special characters with the corresponding strings. The sentence parser detects the sentence boundary in the bio text data using the sentence parser, and the part-of-speech tagger then annotates each token in a sentence with POS tags based on their context.

As in the co-training algorithm shown in Tab. 1, redundantly sufficient feature sets (F_1, F_2), which show one example from two different views as the co-training assumption, need to be designed and extracted. We set one

of the feature sets, F_1, as an orthographic feature that has been utilized as the best indicator in bio NER. There are many studies based on orthographic features [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15] and a good one can be found in [7]. Another set of features, F_2, is extracted as a context based feature with lexicons that provide domain knowledge in a consistent manner. Here, we use the lexicons prepared by the authors [18]. The outputs of the feature processing schema, the feature sets from orthographic, and lexicons are shown in Fig. 2. The regex strings, which reveal orthographic information, are matched to each token to give orthographic information to the output, where "t" means matched and "f" means unmatched. Analogous to the extraction of orthographic features, the context features are extracted; however, the candidates, which can contain more than one token, are matched through the lexicons.

In the feature processing, since we deal with individual feature sets as shown in Fig. 2, the feature conversion, which is shown in Tab 2, 3 and converts three-class labeled examples to two-class labeled examples, is preformed whenever co-training refreshes the labeled data (L) for training in each round. Since the extraction of both orthographic and context feature sets relies on words from text data, the word feature representation is obtained in two different ways as shown in the algorithm, where the word representation for context feature set is built by merging the tokens of an example with the orthographic features if the example is labeled as "TRUE" and the word representation for orthographic feature set is attained by tokenizing the candidate word feature of an example with context features. At the end of the conversion of context to orthographic features, we filter "B" and "I" examples labeled as "O" using BIO classifier, which learned via the data set of orthographic features, because the candidates can consist of three examples of "B", "I" and "O".

C. *Settling the classifiers of co-training*

The classification model (C_1) classifies an example into BIO label format based on the orthographic features, while the other one (C_2) classifies candidates into either true or false based on context based features. In Fig. 2, the right most columns of the datasets are class labels and the dataset shown on the left side of Fig. 2 consists of tokens labeled with "B", "I" or "O"; however, the dataset shown on the right side of Fig. 2 consists of named entity candidates labeled with "TRUE" or "FALSE". The two classifiers supply one another with the batch of new training examples, which are predicted as most confident, from unlabeled data in each round of co-training by using the feature processing schema.

Once the classification models have been built, we combine them in order to further improve the performance of our previous NER method [15]. Since the candidate classifier accuracy has been reported as being higher and the classifier solves the boundary problem, the combined classifier completely accepts the classification result of the candidate classifier, and in addition to this result, it also takes BI examples labeled by the BIO classifier. In other words, the BI examples that are not included in the true candidates resulted by the candidate classifier could belong to the discarded named entities and are viewed as complement to the true candidate examples that can be handled by BIO classifier, which covers the whole token in the text data into the classification process.

III. EXPERIMENTAL RESULTS

A. *Labeled and unlabeled data*

To conduct experiments to show the efficiency of the proposed integration approach, we used GENIA v3.02 corpus [19], in which the bio named entities are annotated with their semantic label. The GENIA corpus consists of 2,000 MEDLINE abstracts, 18,546 sentences, 400,000 words, and 528,113 tokens. Note that the number of sentences and tokens is derived by the Stanford NLP tool [20].

254,139 of PUBMED abstracts, which contain gene and protein keywords gathered since 2009, were collected as unlabeled data for semisupervised learning. The abstracts are stored in one-line-abstract format so that the co-training algorithm randomly selects a number within 254,139, which is the line number corresponding to an abstract.

B. *Experimental setup*

The parameter selection (L, u, n, p in Tab. 1) for co-training is the one important consideration in practice. We started with as minimum labeled data as possible for the initial seed of co-training to verify the effectiveness of the

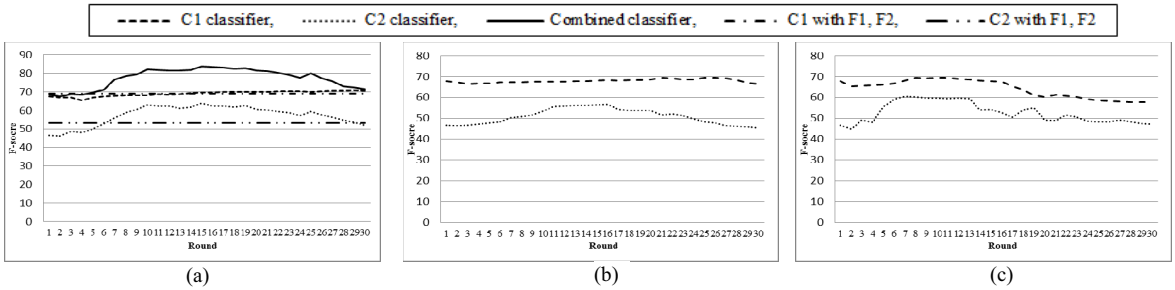


Figure 3. Variation of F-score values in varying rounds of co-training. In (a), F-scores of C_1, C_2 classifiers learned only on labeled data using both F_1, F_2 feature sets are compared against that of the classifiers from co-training. (b), (a) and (c) show results of the basic classifiers when the number of picked examples in each round are n=2,000 p=6,000; n=4,000 p=12,000 and n=6,000 p=18,000, respectively.

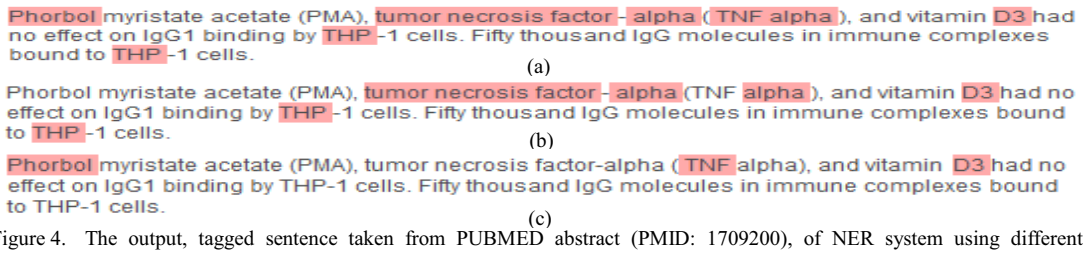


Figure 4. The output, tagged sentence taken from PUBMED abstract (PMID: 1709200), of NER system using different classifiers. Named entities are highlighted. (a), (b) and (c) are outputs from the combined classifier, C_1 classifier with F_1, F_2 feature sets and C_2 classifier with F_1, F_2 feature sets, respectively.

integration method. Hence, we randomly select 500 abstracts from the GENIA corpus as the initial labeled seed (L), while remainder of the GENIA corpus is used as the evaluation dataset for the classifiers. 1,900 abstracts from the unlabeled data (U) are randomly picked for the unlabeled subset (u). Each round which is replenished by randomly selected 500 abstracts and from which the most confident 4,000 "TRUE" examples (n) and 12,000 "FALSE" examples (p) are picked by each classifier (C_1, C_2) for the training dataset. The influence of these parameters on training efficiency was finally reported in our experiment.

We should note that in the original co-training algorithm, the parameter values are given as the number of examples of the classification problem. In our case, however, the values are given as the number of abstracts, each of which is approximately composed of 160 candidates, to avoid the complexity caused by a parameter metric.

The classifiers in co-training could be any of the classification algorithms. However, during the co-training process, many rounds need be conducted, resulting in very long time spent building the classification models. In our experiment, we tried to use SVM, but we faced the time complexity problem. We then select Bayesian classifier for both classifiers of co-training since it takes a short time to train, and the output is a confident score indicating how well an example belongs to the classes.

C. *Performance evaluation*

We ran the co-training until no improvement was observed in its round. Once the classifiers were modified by learning the training dataset in each round, the classification models were stored and evaluated on the test dataset. The classifier evaluation is done individually, and the classifiers are then combined in order to investigate the training advancement in the round.

Fig. 3 shows the plots of classification F-score versus the number of rounds of the co-training for the three different runs, where the number of examples to be picked for the training data is given various values. In Fig. 3a, we compare the F-score values of the classifiers from co-training with the results of the classifiers learned only on the labeled dataset, using both feature sets. The classifiers from co-training beat the classifiers with both feature sets as the co-training round is increased. The notable result of the experiment is that the combined classifier, F-score that is shown as a solid line and achieved 83.6% in the 15th round, gained significant advantage from the unlabeled data.

As shown in Figs. 3a, 3b, and 3c, the F-score values of the classifiers C_1 and C_2 are different. The F-score in Fig. 3b is increased slowly, while the F-score shown in Fig. 3c is increased in a few rounds; however, both did not reach high value compared with F-score in Fig. 3a, the base run. Our experiment, therefore, shows that the learning advancement to be gained from unlabeled data using co-training depends on the number of examples picked for training data in each round of co-training.

Fig. 4 shows the output results of NER system using different classifiers showing the best performance. The combined classifier recognizes the bio named entity better than the other two classifiers.

Note that during the experiment, we ran the co-training over various parameter values and with different numbers of co-training rounds, starting from as low as 2 and going up to 200 to determine values that give the best results. The results of those runs are summarized in this section.

D. *Discussion and open question*

From the experimental results, we observed interesting issues of co-training to discuss. As we have seen in Fig. 3, the learning advancement from one round to the next round is different when parameters n and p are different. The large number of examples to be picked improves classifier F-score in a few rounds, but the improvement no longer exists and is inconsistent. When the number of examples to be picked is too small, the improvement of classifier F-score is also small even if it is more stable than that of the previous case. Therefore, the large number of examples to be picked increases the probability under which noise corresponding to the examples is selected as the correct example, and this fact degrades the improvement in future rounds. On the other hand, the small number of examples to be picked decreases the probability under which the informative example is selected for the training dataset, and this fact reduces improvement in future rounds. In the experiment, we specify the run shown in Fig. 3a as the base run, which achieves a better result among the results from the various runs.

In the base run shown Fig. 3a, the improvement of classifier F-score no longer exists even if it reaches the highest value among the different runs. In many other cases, co-training advances are made in improving F-score early, and it then the improvement of F-score is reduced or

completely diminished as the round goes up. We think this happens because there has not been an informativeness control in co-training. It means that during the learning of co-training, it is possible to pick the same informative examples in different rounds, since co-training elects the examples according to their confidence scores of classification, while some noise can be embedded in the examples to training datasets.

Finally, we got two important questions that need to be considered in practice. Can we formulate the co-training parameters in a particular domain, where the example distribution as well as other character is known, to achieve an accurate result? And how can we modify the co-training algorithm with an informativeness controller tool that picks an informative example for the training dataset in every different round?

IV. CONCLUSION

We presented a semisupervised learning approach to cover knowledge from unlabeled data to recognize bio named entities in text. An integration method has been described to combine completely different classifiers in co-training fashion. To do so, we modified the original co-training algorithm with a scalable feature processing schema, which extracts the bio NER feature from a number of unlabeled data and converts different types of feature sets, and a noise filter.

To verify the effectiveness of the integration methodology, we run the co-training with a small number of labeled datasets in different environments where the parameters and round of co-training are varied, and the base result shows that the classifiers of the co-training gain significant learning improvement from unlabeled data.

While we examined the results, we found out that the parameters of co-training directly affect the learning progress in and that the improvement of classifier F-score no longer exists, because noise can be elected as training example and that there has not been an informativeness control in co-training.

ACKNOWLEDGMENT

This work was supported by the the grant of the Korean Ministry of Education, Science and Technology (The Regional Core Research Program/Chungbuk BIT Research-Oriented University Consortium), National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. 2011-0001044), and Korea Biobank project (4851-307) of the Korea Centers for Disease Control and Prevention.

REFERENCES

[1] H. Dai, Y. Chang, R. T. Tsai, and W. Hsu, "New Challenges for Biological Text-Mining in the Next Decade," Journal of computer science and technology, 25(1): 169, 2010.

[2] L.J. Gong, and X. Sun, "ATRMiner: A system for Automatic Biomedical Named Entities Recognition," ICNC 2010, pp. 3842-3845, 2010.

[3] S. Zhao, "Named Entity Recognition in Biomedical Texts using an HMM Model," Proceedings of the COLING 2004 International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA), 2004.

[4] Z. GuoDong, S. Jian, N. Collier, P. Ruch, and A. Nazarenko, "Exploring Deep Knowledge Resources in Biomedical Name Recognition," COLING 2004 International Joint workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP), pp. 99-102, 2004.

[5] K. M. Park, S. H. Kim, D. G. Lee and H. C. Rim, "Boosting Lexical Knowledge for Biomedical Named Entity Recognition," Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004), pp. 7599, 2004.

[6] T. Mitsumori, S. Fation, M. Murata, K. Doi, and H. Doi, "Gene/protein name recognition based on support vector machine using dictionary as features," BMC Bioinformatics, 2005.

[7] N. Collier, and K. Takeuchi, "Comparison of character-level and part of speech features for name recognition in biomedical texts," Journal of Biomedical Informatics, pp. 423-435, 2004

[8] Z. Ju, J. Wang, and F. Zhu, "Named Entity Recognition From Biomedical Text Using SVM," Bioinformatics and Biomedical Engineering (iCBBE 2011), 2011.

[9] J. Finkel, S. Dingare, H. Nguyen, M. Nissim, C. Manning, and G. Sinclair, "Exploiting Context for Biomedical Entity Recognition: From Syntax to the Web," Joint Workshop on Natural Language Processing in Biomedicine and Its Applications at Coling 2004, 2004.

[10] B. Settles, "Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets," Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004), 2004.

[11] S. Chan, and W. Lam, "Efficient Methods for Biomedical Named Entity Recognition," Bioinformatics and Bioengineering, 2007.

[12] C. Hsu, Y. Chang, C. Kuo, Y. Lin, H. Huang, and I. Chung, "Integrating high dimensional bi-directional parsing models for gene mention tagging," Bioinformatics, 2008.

[13] Y. Li, H Lin, and Z. Yang, "Integrating rich background knowledge for gene named entity classification and recognition," BMC Bioinformatics, 2009.

[14] L. Yang, and Y. Zhou, "Two-phase Biomedical Named Entity Recognition based on Semi-CRFs," Bio-inspired Computing: Theories and Applications (BIC-TA), 2010.

[15] T. Munkhdalai, M. Li, E. Namsrai, O. Namsrai, and K. H. Ruy, "BFSM: Finite State Machine Learned as Name Boundary Definer for Bio Named Entity Recognition," ICAST 2011, 2011.

[16] A. Blum, and T. Mitchell, "Combining Labeled Data with Co-Training," 11th Annual Conference Computational Learning Theory, 1998.

[17] K. Nigam, and R. Ghani, "Analyzing the Effectiveness and Applicability of Co-training," Information and Knowledge Management, 2000.

[18] L. Tanable, and J. Wilbur, "Tagging Gene and Protein names in Full Text articles," Workshop on Natural language processing in the Biomedical Domain, 2002.

[19] J.D. Kim, T. Ohta, Y. Tateishi, and J. Tsujii, "GENIA corpus-a semantically annotated corpus for bio-text mining," Bioinformatics 2003, 19(Suppl. 1):18-2, 2003.

[20] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network," Proc. of HLT-NAACL 2003, pp. 252-259, 2003.